Department of Computer
CMPT 250.6
Midterm Exam #2
CLOSED BOOK

Time: 50 Minutes

Marks

(10) 1. Use mathematical induction to prove that

$$3n^5 + 5n^3 + 7n$$

is divisible by 15 for each nonnegative integer.

(8) 2. Consider a hash table where open addressing and linear probing are used for collision resolution. Assume that for a new item, the hash function is equally likely to yield each of the locations of the table. Also assume that the table has 9 locations numbered 1 through 9, and locations 2, 3, 5, 7, 8, and 9 are presently occupied. For each unoccupied location of the table (1, 4, 6), what is the probability that the insertion of a new item will result in the item being placed in that location.
Show your work.

(8) 3. One way to handle deletions in a lexically-ordered binary tree is to have each node contain a Boolean flag attribute that is set to True, if the node has been marked for deletion. When over half of the nodes in the tree have been marked for deletion, the tree is reorganized so that all nodes whose delete flags have a value of True are removed (i.e., deleted).

For this question assume the nodes are ordered by an integer key.

a) Define a new class for the above type of tree that inherits from class LINKED_SIMPLE_TREE_UOS[G] which contains a Boolean delete flag.

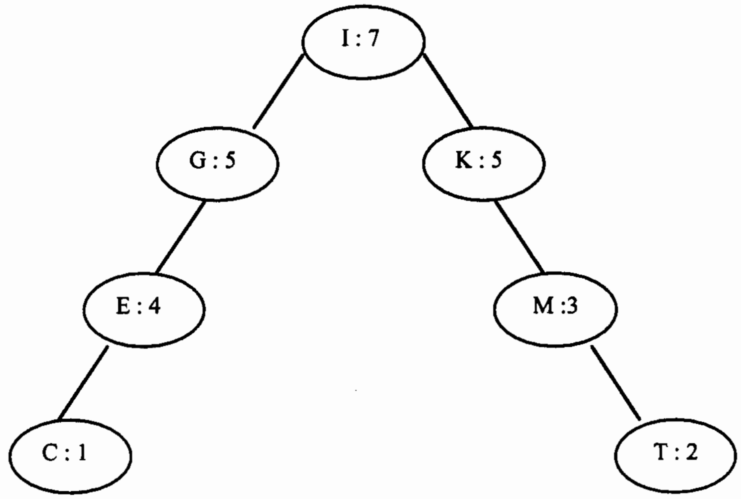
b) Write a recursive function *delete_test*, as a new feature of the new class in part a) which when applied to a given tree object, returns True if more than one half of the tree's nodes have their delete flags set to True.

NOTE: You ARE NOT TO WRITE the procedure that actually reorganizes, i.e., deletes all nodes in a given tree whose delete flags have been set to True.

Some of the features of LINKED_SIMPLE_TREE_UOS[G] are:

is_empty : BOOLEAN
root_left_subtree : LINKED_SIMPLE_TREE_UOS[G]
root_right_subtree : LINKED_SIMPLE_TREE_UOS[G]
root_item[G]

(14) 4. Consider the following weight-balanced tree:



i) What is the expected length of search for this tree?

ii) Draw the height-balanced tree that is created by inserting the nodes in an initially empty tree according to a postorder traversal of the given weight-balanced tree.

iii) Assuming the same relative frequencies of access as given in the weight-balanced tree, what is the expected average length of search for the tree you constructed in part ii)?

iv) Draw the weight-balanced tree that would result if node 'E' is accessed twice and then node 'I' is deleted from the weight-balanced tree given above.

(10) 5. A hypothetical insurance premium program computes annual car insurance premiums based on two parameters: the policyholder's age and driving record as follows —

premium := base_rate * *age_multiplier – safe_driving_reduction*

The *age_multiplier* is a function of the policyholder's age. The *safe_driving_reduction* is given when the *current_points* (assigned by traffic courts for moving violations) on the policyholder's driver licence are below an age-related cutoff. The *base_rate* changes from time to time; it is currently $1000 for an annual premium.
The following table is used to compute premiums:

age_range	age_multiplier	points_cutoff	safe_driving_reduction
$16 \le$ age < 25	2.8	6	$50
$25 \le$ age < 50	1.0	3	$100
$50 \le$ age < 90	1.5	9	$150

For example, for a policyholder of age 70 with a *current_points* value of 6, the premium is: $1000 * 1.5 – 150 = $1350
The input in this example is a follows:
age = 70 *current_points* = 6
Using an equivalence class partitioning approach, formulate equivalence classes for this program. NO TEST CASES ARE REQUIRED.